

Viruses leave *fingerprints* when they infect an application, your System file or any other file. In technical jargon, these fingerprints are called **resources**. What **VirusDetective™** does is provide a means to search for these resources using various matching criteria. Before I go further, I should mention that the intention of **VirusDetective™** is to *detect* viruses, NOT to remove them. Unless you *absolutely* know what you are doing, you should *not* use the **Remove** button but either replace the infected file or use one of the various repair programs available (several are distributed on the **VirusDetective™** disk. Also on this diskette are several pieces of documentation about the various viruses from different sources.) Other sources of documentation include the November 1988 issue of MacWorld.

As of this writing, the known viruses in the Mac community are Peace, nVIR (with variants Hpat, AIDS and MEV#), Scores, INIT29 and ANTI. When these viruses infect your Mac, they sometimes infect your System file in a different manner than how they infect other files. As shipped, **VirusDetective™ 3.0** (and later) is configured to look for the non-System and System file resources. You may want to remove the System only search patterns (after saving them to a file) once you have checked your System file.

The search strings (for **VirusDetective™ 3.0** and later only) are:

**Resource Start & Size<800 & Data 2F3A#F00#C80**  
; For finding nVIR, etc. (Appl's/Finder)  
**Filetype=ZSYS & Resource INIT & Size<800 & Data 2F3A#F00#C80**  
; For finding nVIR, etc. (System)  
**Resource Start & Size<800 & Data 41FA#92E#797**  
; For finding INIT29 in Appl's/Finder  
**Filetype≠APPL & Resource INIT & Size<800 & Data 41FA#92E#797**  
; For finding INIT29 in non-Appl's  
**Resource Start & Size<8000 & Data FD38#FBA#5A3**  
; For finding Scores in Appl's/Finder  
**Filetype≠APPL & Resource INIT & Size<1100 & Data FD38#FBA#5A3**  
; For finding Scores in System, etc.  
**Resource Start & Pos -1344 & Data 060CA9#643E9** ; For finding ANTI  
**Resource INIT & Name "RR"** ; For finding Peace

The two **nVIR** searches are conclusive that you have been infected by these viruses. The trademark of these viruses is that they add **nVIR**, **Hpat**, **AIDS**, and **MEV#** resources to your file as well. This search pattern was designed to find *active* **nVIR** viruses. You can use **Resource nVIR & Any** (add like searches for **Hpat**, **AIDS**, and **MEV#**) if you want to check for inactive infections as well.

The two **Scores** searches will find *active* **Scores** in applications, the System file and the “**Scores**” data file. These searches will also find some mutations of the **Scores** virus that may be created.

**INIT Name "RR"** will find the Peace virus. This virus has probably already deleted itself because it had an expiration date and deleted itself. Thus you probably do not need to keep it among your daily search strings.

The two **INIT29** searches will find *active* **INIT29** in applications, the System file and any other non-application files infected. These searches will also find some mutations of the **INIT29** virus that may be created.

**Resource Start & Pos -1344 & Data 060CA9#643E9** is conclusive evidence you have the “ANTI” virus. “ANTI” does not infect the System file.

The “**Size<xxx**” tests in the above searches were chosen to speed up searching by not testing resources that don’t fit the *currently known* virus size (with a little leeway). However, it is possible for a malicious person to add “garbage” to a virus increasing its size thus causing those tests to fail. Thus, if you really suspect a virus and **VirusDetective™** doesn’t report any, you might want to remove the Size restrictions.

My advice if you suspect you have a virus is first, don’t panic! Next, if you can replace the infected file from your original disks or an uninfected backup, do so. If you can’t, use one of the eradication programs for that virus. There are several good ones in the public/shareware domain. Finally, use **VirusDetective™** to check all new files and disks before you use them. You should also use programs like **Vaccine** or **Gate- Keeper** to help you prevent any new virus from infecting you.

Due to the way some viruses operate, if a file is infected by more than one virus **VirusDetective™** may only find a single virus using the above search strings. Therefore, if you cannot replace the file with a new copy and choose to use one of the disinfecting programs on it, you should **always** check the file again to be sure it is clean.

As any new viruses are discovered and brought to my attention, I will send all my **registered** users notification of new search strings. You should try to keep informed by reading all the timely Macintosh literature or subscribing to a BBS system like Delphi, Compuserve or GENie.

Jeff Shulman  
PO Box 521  
Ridgefield, CT 06877-0521  
USA  
6/14/89

AppleLink: KILROY  
CIS: 76136,667  
Delphi: JEFFS  
GEnie: KILROY  
MA Bell: (203) 792-1521

**VirusDetective™** is shareware. You can register yourself by sending \$35.00 (non-US users \$40.00 in US funds on a US bank) for user license only (no diskette). If you send \$40.00 (non-US users \$45.00 in US funds on a US bank), you will receive a user licence *and* a diskette with the latest version of **VirusDetec- tive™**, further documentation on viruses collected from various sources, other useful PD virus programs collected from various sources and the latest versions of all my other software:

<b>FontDisplay™ Ltd.</b>	a limited demoware version of <b>FontDisplay™</b> , a program to display and print font files.
<b>WriteFontSize™</b>	a shareware DA that allows you to paste <i>any</i> character of any installed font to the clipboard.
<b>DiskLock™</b>	a freeware DA that can write protect any floppy or hard disk.

Multiple copy/site licensing is available. Write or call for details.

## VirusDetective™ 3.0 (and later) Search Pattern Syntax

```
<search-string> :=  
    <file-string> {; Comment}  
    <file-string> & <resource-string> {; Comment}  
    <resource-string> {; Comment}
```

This says a search string is either a <file-string>, a <file-selector> followed by the character ‘&’ followed by a <resource-string> or just a <resource-string>. All of them may be followed by an optional (indicated by the {}’s) ‘;’ and comment string.

<file-string>’s can actually occur anywhere in the search-string, not just at the beginning. In general you want them first to speed the search process.

```
<file-string> :=  
    Creator <op> <CID>  
    Filetype <op> <FID>  
  
<op> :=  
    =  
    ≠  
    >  
    <  
  
<CID> := 4 character file creator  
<FID> := 4 character file type
```

This says a <file-string> is the word “**Creator**” or “**Filetype**” followed by a comparison operator ‘=’ (for “is equal to”), ‘≠’ (for “is not equal to”) (type Option-= to get this character), ‘>’ (for “is greater than”) or ‘<’ (for “is less than”) followed by a four character file creator or filetype. E.g. “**Filetype** = **APPL**” would match all applications.

```
<resource-string> :=  
    <resource-selector> & <resource-comp> [ & <resource-comp>]
```

A <resource-string> is a <resource-selector> followed by one or more (indicated by the []’s) <resource-comp>’s each separated by a ‘&’.

```
<resource-selector> :=  
    Resource Start  
    Resource <RID>  
  
<RID> := 4 character resource type
```

A <resource-selector> is either “**Resource Start**” which means the first executed CODE resource or “**Resource XXXX**” where XXXX is some resource type like “nVIR”. Note: It *must* be exactly 4 characters, including spaces.

```
<resource-comp> :=  
    Any  
    Data <pattern>  
    ID <op> <snum>  
    Name <sep><string><sep>  
    Pos <snum> & Data <pattern>  
    Size <op> <num>  
  
<snum> :=  
    -<num>  
    <num>  
  
<num> := unsigned decimal number  
<sep> := any single character  
<string> := string of up to 255 characters
```

**Any** Matches any <resource-selector> resource. E.g. “**Resource nVIR & Any**” would match any nVIR resource.

**Data** Matches any <resource-selector> resource containing the <pattern> (described below). You can specify an optional starting offset position with the “**Pos**” keyword. Positive offsets add to the beginning and negative offsets subtract from the end. E.g. “**Resource Start & Pos -1344 & Data 060CA9#643E9**” starts searching the first executed CODE resource for that pattern 1344 bytes from the end of it.

The “**Data**” keyword must be the last keyword in a search string. The “**Pos**” keyword (if present) can occur anywhere before the “**Data**” keyword.

- ID** Matches any <resource-selector> resource whose resource ID satisfies the given relationship. E.g. “**Resource CODE & ID > 10**” matches any CODE resource whose ID is greater than 10.
- Name** Matches any <resource-selector> resource whose name is enclosed in the separator characters. E.g. “**Resource INIT & Name "RR"**”.
- Size** Matches any <resource-selector> resource whose resource size satisfies the given relationship. E.g. “**Resource MEV# & Size = 722**” matches any MEV# resource whose size is equal to 722.

```
<pattern> :=  
    <hex-pattern>  
    <ascii-pattern>  
<hex-pattern> :=  
    <hex-byte-word>{<hex-pattern>}  
    #<hex-char>{<hex-pattern>}  
<hex-byte-word> :=  
    <hex-char><hex-char>  
<hex-char> := character 0 through 9 or A through F  
<ascii-pattern> :=  
    "<string>"  
    '<string>'
```

A Data match pattern can be specified as a sequence of hex digits, two per byte, or as a ASCII string enclosed in either single or double quotes. An ASCII pattern must match its entire pattern exactly to be considered “a match”. A hex pattern can “skip” bytes by using the “#” character followed single hex character, 0 through F, to skip 0 through 15 bytes. E.g. pattern 3C#500 would match a resource containing 3C12C9006A8000.

Spaces may be used between search-string parts to improve readability.

## Edit History

<u>Version</u>	<u>Date</u>	<u>Comments</u>
1.0	4/4/88	Initial version
1.1	4/22/88	Fixed bug where ID/Name searches didn't work in the current System file. Remove search string button didn't unhighlight after removal. Added Filetype and Creator searching.
1.2	5/22/88	Ability to save a log file added. If you typed an edit command in the help dialogs, you got a crash.
2.0	1/14/88	Remember DA location. Cmd-W to close. Added batch disk checking option (also works in the background.) Moved all configuration buttons to main window. Changed "Check Current Folder" to "Check Entire Disk". In the Modifications dialog you can copy the contents of the selected cell. You can now have comments in the search string patterns. You can now keep a single log file for every checked disk. If checking for a file type or creator you can delete the matched file. File/Resource Information dialog added (also allows you to change some resource attributes.) You can copy the name and address from the About dialog to the clipboard. The Continue button is the default when a search matches. Rearranged main dialog. Added single file searching. Added Unattended foreground option. Added option to put Get Info stuff in log file. Added "Dual" search pattern. OK's changed to Save's in Option/Mod dialogs. Fixed searching of currently opened files. Delete/Remove button now labelled Delete or Remove based on search match. Beep if any illegal Command key character is entered. Don't allow Cmd-D/Cmd-R to be interchanged for Delete/Remove. Beeps if you click outside the content region if it is waiting for Continue/Cancel/Remove/Info. Also button clicks don't activate immediately anymore (happened under certain circumstances.) Switched Scan buttons with Credit/Help by popular demand. Added/modified modal dialog/alert hooks to repaint frames default button. Really fix immediately activating buttons. Ignore autokey events in dialogs. Beef up resource removal warning. Make sure it is us to be updated before we do anything in the dialog hooks. Don't allow Cmd key chars in genericFilter. Only accept Return on the get directory dialog iff control is highlighted. Resource attributes weren't being written to log file. Clip file name and search strings to their rectangles. Added JStart search.
2.1	2/25/89	Converted to MPW 3.0. Added JData and Data searches. Save open files at check time, not opening. Warn user if opening resource fork still fails after read only. Don't allow resource change/remove attempts on read-only files. Auto floppy check would skip floppies and/or check twice under certain circumstances; fixed. Close all opened read-only resource forks. Make sure any system fonts aren't deleted when closing a resource file containing them. On auto-check do only new floppies. Don't alert user if running in unattended mode.
2.1.1	2/28/89	Remove button didn't highlight. Would allow certain illegal data search patterns.
2.2	3/4/89	Maintain C and Pascal string versions of our file name. Put back search current folder button.
2.2.1	3/31/89	Move SetVol's to right before file checking. Fixes problem with SuperSpool.
3.0	5/20/89	Revamped built-in Help system for each major dialog. Help text can now be copied to the clipboard for saving/printing. Can now read/write search strings from/to a file. File modification date now always reported in Info dialogs and logged. File deletion available on any match. Can now skip a

matched search string or file without aborting entire scan.  
Totally restructured search string language which allows for  
multiple search criteria per string for maximum flexibility,  
effectiveness and speed.

3.0.1

6/16/89

Fixed bug where it missed System file resources under ID=x tests  
(HomeResFile bug (again!)). If first Data scan failed, restore offset for  
future scans so they don't always fail.